Duan Nguyen

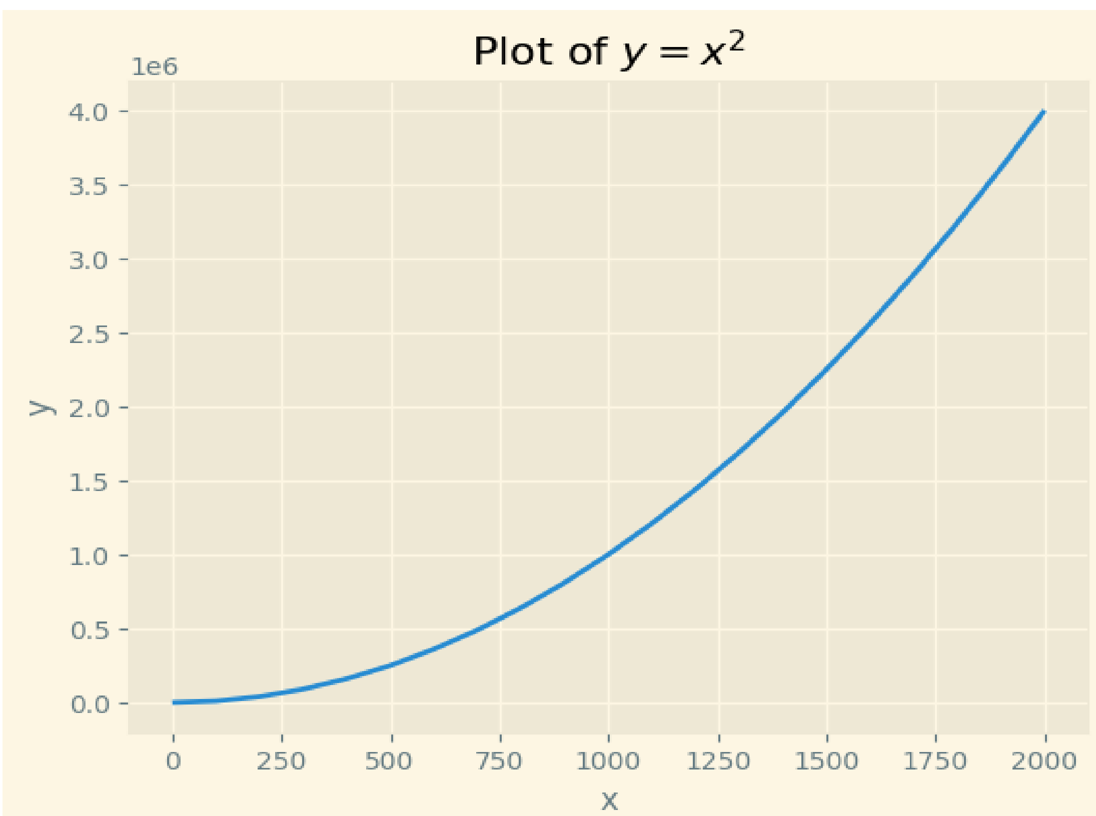
CSCI 406

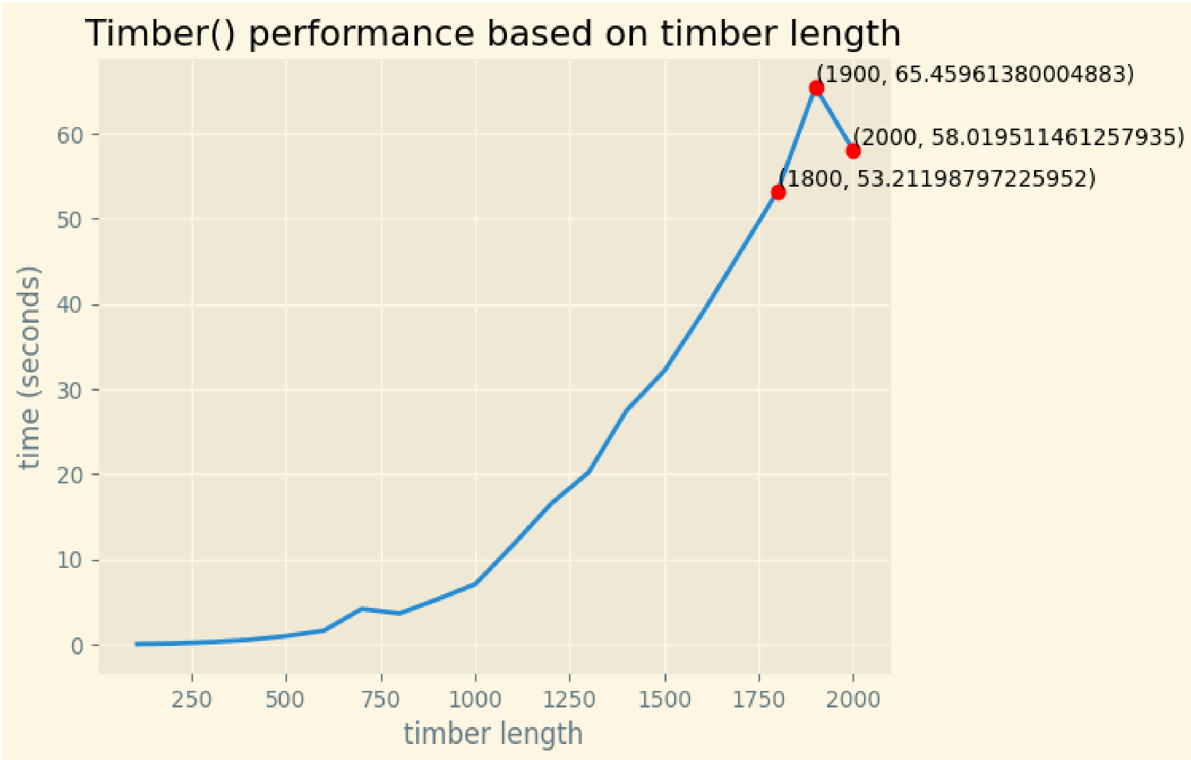
10th April 2024

Timber Problem – Analysis – Part 2

For the Bottom-Up implementation of the timber problem, each cell will depend on the cell below it and the cell on its left, so the table would be filled diagonally. Within the function $T(i, j)$, the first outer loop will be moving through the column and stop at the length of the log. The second inner loop will be moving through both the column and the row starting from where the column pointer of the first loop, but the second loop will also stop at the length of the log. In theory, the function will loop through **n**-by-**n** times if the length of log is **n**, which mean that the average time complexity of the function $T(i, j)$ is $\Theta(n^2)$.

To verify the theory, two graphs were made using the range of 0 to 2000 by step of 100, one is $y(x) = x^2$ and the other is the recorded performance of the Bottom-Up implementation of $T(i, j)$.





In the recorded data, the trend is almost like the function $y(x) = x^2$ but at different scale if not for the outlier at $\boldsymbol{n} = 2000$. Using the other two highlighted datapoint, square root of the measured time of $\boldsymbol{n} = 1900$ is around 8.09, and the square root of the measured time of $\boldsymbol{n} = 1800$ is around 7.29, which means the measured time is increasing quadratically at a very large **n**. Therefore, the asymptotic complexity of the Bottom-Up implementation of $T(i, j)$ is $\Theta(x^2)$.